

What is claimed is:

sub AI

1. A processor for reading instructions from a memory according to a program counter, the memory storing instructions in one-byte units, and for executing the read instructions,

the program counter including a first program counter and a second program counter,

the first program counter indicating a storage position of a processing packet in the memory, the processing packet being composed of an integer number of the one-byte units,

the second program counter indicating a position of processing target instruction in the processing packet, the processing target instruction being an operation to be executed by the processor.

sub C

2. The processor of Claim 1, including a first program counter updating means and a second program counter updating means,

the second program counter updating means incrementing a value of the second program counter in accordance with an amount of instructions that were executed in a preceding cycle and sending any carry generated in an incrementing to the first program counter updating means, and

the first program counter updating means adding the carry received from the second program counter updating

Sub C' 7

12 means to the value of the first program counter.

1 3. The processor of Claim 2, further including:

2 program counter relative value extracting means for
3 extracting, when an instruction being executed includes a
4 program counter relative value that is based on an address
5 of a first instruction executed in a present cycle, the
6 program counter relative value; and

7 calculating means for adding the program counter
8 relative value to the value of the first program counter
9 and the value of the second program counter, and setting an
10 addition result as the value of the first program counter
11 and the value of the second program counter.

1 4. The processor of Claim 3,

2 wherein the calculating means includes a first
3 calculating unit and a second calculating unit,

4 the second calculating unit adding the value of the
5 second program counter and lower bits of the program
6 counter relative value, setting a result of an addition as
7 the value of the second program counter, and sending any
8 carry generated in the addition to the first calculating
9 unit,

10 the first calculating unit adding the value of the
11 first program counter, upper bits of the program counter
12 relative value, and any carry received from the second
13 calculating unit, and setting a result of an addition as

Sub C¹ 7

14 the value of the first program counter.

1 5. The processor of Claim 3,

2 wherein the calculating means includes a first
3 calculating unit and a second calculating unit,

4 the second calculating unit adding the value of the
5 second program counter and lower bits of the program
6 counter relative value without generating a carry, and
7 setting a result of an addition as the value of the second
8 program counter,

9 the first calculating unit adding the value of the
10 first program counter and upper bits of the program counter
11 relative value, and setting a result of an addition as the
12 value of the first program counter.

1 6. The program counter of Claim 3,

2 wherein the calculating means adds the value of the
3 first program counter and upper bits of the program counter
4 relative value, sets a result of an addition as the value
5 of the first program counter, and sets lower bits of the
6 program counter relative value as the value of the second
7 program counter.

1 7. The processor of Claim 3,

2 wherein the calculating means adds the program
3 counter relative value and a value whose upper bits are the
4 value of the first program counter and lower bits are the

Sub C' 7

5 value of the second program counter, and sets upper bits of
6 a result of an addition as the value of the first program
7 counter and lower bits of the result as the second program
8 counter.

1 8. The processor of Claim 2, further including:

2 program counter relative value extracting means for
3 extracting, when an executed instruction includes a program
4 counter relative value that is based on an address of the
5 executed instruction, the program counter relative value;

6 program counter amending means for amending the value
7 of the first program counter and the value of the second
8 program counter to indicate an address of the executed
9 instruction; and

10 calculating means for adding the program counter
11 relative value, the value of the first program counter, and
12 the value of the second program counter, and setting a
13 result of an addition as the value of the first program
14 counter and the value of the second program counter.

1 9. The processor of Claim 2, further including:

2 program counter relative value calculating
3 instruction decoding means for decoding a program counter
4 relative value calculating instruction that performs an
5 addition using a program counter relative value and one of

6 (a) a value of the program counter stored in a
7 register, and

Sub C⁷

8 (b) the value of the first program counter and the
9 value of the second program counter;
10 calculating means for performing the addition
11 indicated by the program counter relative value calculating
12 instruction to generate an addition result; and
13 program counter value updating means for storing the
14 addition result in one of
15 (a) the register, and
16 (b) the first program counter and the second program
17 counter.

SECRET

Sub C²

1 10. The processor of Claim 1,

2 wherein the first program counter indicates a memory
3 address, the memory address being a storage position in the
4 memory of a processing packet that is given by bit shifting
5 the value in the first program counter by $\log_2 n$ bits in a
6 leftward direction, n being a length of a processing packet
7 in bytes.

1 11. The processor of Claim 10, further including

2 an instruction buffer for temporarily storing
3 instructions; and

4 instruction reading means for transferring
5 instructions with a minimum transfer size of one one-byte
6 unit from the memory to the instruction buffer, in
7 accordance with available space in the instruction buffer
8 but regardless of a size of a processing packet.

1 12. An instruction sequence optimizing apparatus, for
2 generating optimized code from an instruction sequence,
3 comprising:

4 address assigning means for estimating a size of each
5 instruction in the instruction sequence and assigning an
6 address to each instruction, upper bits of each address
7 indicating a memory address at which a processing packet is
8 stored and lower bits of each address indicating a
9 processing target instruction in the processing packet;

10 label detecting means

11 (1) for detecting a label, which should be resolved
12 by an address of a specified instruction, from the
13 instruction sequence, and obtaining the address of
14 the specified instruction, and

15 (2) for detecting a label, which should be resolved
16 by a difference in addresses of two specified
17 instructions, from the instruction sequence, and
18 obtaining the addresses of the two specified
19 instructions;

20 program counter relative value calculating means for
21 calculating, when a label which should be resolved by a
22 difference in addresses of two specified instructions has
23 been detected, a program counter relative value by
24 subtracting an address of one of the two specified
25 instructions from an address of another of the two
26 specified instructions;

10 subtraction as lower bits of the program counter relative
11 value, and sending any carry generated in the subtraction
12 to the upper bit subtracting unit, and

13 the upper bit subtracting unit subtracting upper bits
14 of the address of one of the two specified instructions and
15 any carry received from the lower bit subtracting unit from
16 upper bits of the address of the other of the two specified
17 instructions, and for setting a result of a subtraction as
18 upper bits of the program counter relative value.

1 14. The instruction sequence optimizing apparatus of Claim
2 12,

3 wherein the program counter relative value
4 calculating means includes a lower bit subtracting unit and
5 an upper bit subtracting unit,

6 the lower bit subtracting unit subtracting lower bits
7 of the address of one of the two specified instructions
8 from lower bits of the address of the other of the two
9 specified instructions without generating a carry and
10 setting a result of a subtraction as lower bits of the
11 program counter relative value, and

12 the upper bit subtracting unit subtracting upper bits
13 of the address of one of the two specified instructions
14 from upper bits of the address of the other of the two
15 specified instructions, and for setting a result of a
16 subtraction as upper bits of the program counter relative
17 value.

1 15. The instruction sequence optimizing apparatus of Claim
2 12,

3 wherein the program counter relative value
4 calculating means subtracts upper bits of an address of one
5 of the two specified instructions from upper bits of an
6 address of the other of the two specified instructions,
7 sets a result of a subtraction as upper bits of the program
8 counter relative value, and sets lower bits of the other of
9 the two specified instructions as lower bits of the program
10 counter relative value.

1 16. An assembler that generates relocatable code from an
2 instruction sequence, each address of an instruction in the
3 instruction sequence having upper bits that indicate a
4 memory address at which a processing packet is stored and
5 lower bits that indicate a position of processing target
6 instruction that is included in the processing packet,

7 the assembler comprising:

8 label detecting means for detecting a label in the
9 instruction sequence that should be resolved by a
10 difference in addresses between two specified instructions,
11 and obtaining the addresses of the two specified
12 instructions;

13 program counter relative value calculating means for
14 calculating a program counter relative value by subtracting
15 an address of one of the two specified instructions from an

16 address of another of the two specified instructions; and
17 replacing means for replacing the label with the
18 program counter relative value calculated by the program
19 counter relative value calculating means.

1 17. The assembler of Claim 16,

2 wherein the program counter relative value
3 calculating means includes a lower bit subtracting unit and
4 an upper bit subtracting unit,

5 the lower bit subtracting unit subtracting lower bits
6 of the address of the one of the two specified instructions
7 from lower bits of the address of the other of the two
8 specified instructions, for setting a result of a
9 subtraction as lower bits of the program counter relative
10 value, and sending any carry generated in the subtraction
11 to the upper bit subtracting unit, and

12 the upper bit subtracting unit subtracting upper bits
13 of the address of one of the two specified instructions and
14 any carry received from the lower bit subtracting unit from
15 upper bits of the address of the other of the two specified
16 instructions, and for setting a result of a subtraction as
17 upper bits of the program counter relative value.

1 18. The assembler of Claim 16,

2 wherein the program counter relative value
3 calculating means includes a lower bit subtracting unit and
4 an upper bit subtracting unit,

5 the lower bit subtracting unit subtracting lower bits
6 of the address of one of the two specified instructions
7 from lower bits of the address of the other of the two
8 specified instructions without generating a carry and
9 setting a result of a subtraction as lower bits of the
10 program counter relative value, and

11 the upper bit subtracting unit subtracting upper bits
12 of the address of one of the two specified instructions
13 from upper bits of the address of the other of the two
14 specified instructions, and for setting a result of a
15 subtraction as upper bits of the program counter relative
16 value.

1 19. The assembler of Claim 16,

2 wherein the program counter relative value
3 calculating means subtracts upper bits of an address of one
4 of the two specified instructions from upper bits of an
5 address of the other of the two specified instructions,
6 sets a result of a subtraction as upper bits of the program
7 counter relative value, and sets lower bits of the other of
8 the two specified instructions as lower bits of the program
9 counter relative value.

1 20. A linker that generates object code by combining
2 relocatable code, each address of an instruction in the
3 relocatable code having upper bits that indicate a memory
4 address at which a processing packet is stored and lower

5 bits that indicate a position of processing target
6 instruction that is included in the processing packet,
7 the linker comprising:

8 relocation information detecting means for detecting
9 a label in the relocatable code that should be resolved by
10 a difference in addresses between two specified
11 instructions, and obtaining the addresses of the two
12 specified instructions;

13 program counter relative value calculating means for
14 calculating a program counter relative value by subtracting
15 an address of one of the two specified instructions from an
16 address of another of the two specified instructions; and

17 replacing means for replacing the label with the
18 program counter relative value calculated by the program
19 counter relative value calculating means.

1 21. The linker of Claim 20,

2 wherein the program counter relative value
3 calculating means includes a lower bit subtracting unit and
4 an upper bit subtracting unit,

5 the lower bit subtracting unit subtracting lower bits
6 of the address of the one of the two specified instructions
7 from lower bits of the address of the other of the two
8 specified instructions, for setting a result of a
9 subtraction as lower bits of the program counter relative
10 value, and sending any carry generated in the subtraction
11 to the upper bit subtracting unit, and

12 the upper bit subtracting unit subtracting upper bits
13 of the address of one of the two specified instructions and
14 any carry received from the lower bit subtracting unit from
15 upper bits of the address of the other of the two specified
16 instructions, and for setting a result of a subtraction as
17 upper bits of the program counter relative value.

1 22. The linker of Claim 20,

2 wherein the program counter relative value
3 calculating means includes a lower bit subtracting unit and
4 an upper bit subtracting unit,

5 the lower bit subtracting unit subtracting lower bits
6 of the address of one of the two specified instructions
7 from lower bits of the address of the other of the two
8 specified instructions without generating a carry and
9 setting a result of a subtraction as lower bits of the
10 program counter relative value, and

11 the upper bit subtracting unit subtracting upper bits
12 of the address of one of the two specified instructions
13 from upper bits of the address of the other of the two
14 specified instructions, and for setting a result of a
15 subtraction as upper bits of the program counter relative
16 value.

1 23. The linker of Claim 20,

2 wherein the program counter relative value
3 calculating means subtracts upper bits of an address of one

4 of the two specified instructions from upper bits of an
5 address of the other of the two specified instructions,
6 sets a result of a subtraction as upper bits of the program
7 counter relative value, and sets lower bits of the other of
8 the two specified instructions as lower bits of the program
9 counter relative value.

1 24. A disassembler that receives an indication of an
2 address of an instruction in object code and outputs an
3 assembler name of the instruction at the indicated address,
4 each address of an instruction in the object code having
5 upper bits that indicate a memory address at which a
6 processing packet is stored and lower bits that indicate a
7 position of processing target instruction that is included
8 in the processing packet,

9 the disassembler comprising:

10 program counter relative value extracting means for
11 extracting, when the indicated instruction includes a
12 program counter relative value, the program counter
13 relative value from the indicated instruction;

14 label addressing calculating means for adding an
15 address of the indicated instruction to the extracted
16 program counter relative value and setting an addition
17 result as a label address;

18 storing means for storing a label name corresponding
19 to each label address; and

20 searching means for searching the storing means for a

21 label name that corresponds to the calculated label address
22 and outputting the corresponding label name.

1 25. The disassembler of Claim 24,

2 wherein the label address calculating means includes
3 a lower bit calculating unit and an upper bit calculating
4 unit,

5 the lower bit calculating unit for adding lower bits
6 of the address of the indicated instruction and lower bits
7 of the program counter relative value, setting a result of
8 an addition as lower bits of a label address, and sending
9 any carry generated by the addition to the upper bit
10 calculating unit, and

11 the upper bit calculating unit adding upper bits of
12 the address of the indicated instruction, upper bits of the
13 program counter relative value, and any carry received from
14 the lower bit calculating unit, and setting a result of the
15 an addition as upper bits of the label address.

1 26. The disassembler of Claim 24,

2 wherein the label address calculating means includes
3 a lower bit calculating unit and an upper bit calculating
4 unit,

5 the lower bit calculating unit adding lower bits of
6 the address of the indicated instruction and lower bits of
7 the program counter relative value without generating a
8 carry, and setting a result of an addition as lower bits of

14 instruction writing means for writing the replacement
15 instruction into the processing packet in the instruction
16 buffer over an instruction that is indicated by the lower
17 bits of the indicated address; and

18 processing packet writing means for writing the
19 processing packet in the instruction buffer back into the
20 memory after the replacement instruction has been written.

1 29. A compiler that generates an instruction sequence from
2 source code,

3 the compiler generating a program counter relative
4 value calculating instruction that is executed by a
5 processor, the program counter relative value calculating
6 instruction being an instruction that performs a
7 calculation using a first value and a program counter
8 relative value and uses a result of the calculation to
9 update the first value, the first value being one of

10 (a) a value of a program counter stored in a
11 register, and

12 (b) the value stored in a program counter of the
13 processor,

14 wherein upper bits of the first value indicate a
15 memory address at which a processing packet is stored, and
16 lower bits of the first value of the program counter
17 indicate a processing target instruction that is included
18 in the processing packet.

COPIED FROM THE FILE

1 30. The compiler of Claim 29,

2 wherein the processor includes a lower bit
3 calculating unit and an upper bit calculating unit,

4 the program counter relative value calculating
5 instruction having the lower bit calculating unit perform a
6 lower bit calculation and the upper bit calculating unit
7 perform an upper bit calculation,

8 the lower bit calculation being an addition using
9 lower bits of the first value and lower bits of the value
10 of the program counter relative value, where a result of
11 the lower bit calculation is set as the lower bits of the
12 first value and any generated carry is sent to the upper
13 bit calculating unit, and

14 the upper bit calculation being an addition using
15 upper bits of the first value, upper bits of the value of
16 the program counter relative value and any carry received
17 from the lower bit calculating unit, where a result of the
18 upper bit calculation is set as the upper bits of the first
19 value.

1 31. The compiler of Claim 29,

2 wherein the processor includes a lower bit
3 calculating unit and an upper bit calculating unit,

4 the program counter relative value calculating
5 instruction having the lower bit calculating unit perform a
6 lower bit calculation and the upper bit calculating unit
7 perform an upper bit calculation,

2 instruction sequence optimizing program that generates
3 optimized code from an instruction sequence, the
4 instruction sequence optimizing program including:
5 an address assigning step for estimating a size of
6 each instruction in the instruction sequence and assigning
7 an address to each instruction, upper bits of each address
8 indicating a memory address at which a processing packet is
9 stored and lower bits of each address indicating a
10 processing target instruction in the processing packet;
11 a label detecting step (1) for detecting a label,
12 which should be resolved by an address of a specified
13 instruction, from the instruction sequence, and obtaining
14 the address of the specified instruction, and
15 (2) for detecting a label, which should be resolved
16 by a difference in addresses of two specified
17 instructions, from the instruction sequence, and
18 obtaining the addresses of the two specified
19 instructions;
20 a program counter relative value calculating step for
21 calculating, when a label which should be resolved by a
22 difference in addresses of two specified instructions has
23 been detected, a program counter relative value by
24 subtracting an address of one of the two specified
25 instructions from an address of another of the two
26 specified instructions;
27 a converting step
28 (1) for converting an instruction that has a label

29 that should be resolved by an address of a specified
30 instruction into an instruction with a size that is based
31 on a size of the address of the specified instruction,

32 (2) for converting an instruction that has a label
33 that should be resolved by a difference in
34 addresses of two specified instructions into an
35 instruction with a size that is based on a size of
36 the program counter relative value calculated from
37 the addresses of the two specified instructions;
38 and

39 an optimized code generating step for generating
40 optimized code by converting addresses of instructions in
41 accordance with the sizes of instructions after conversion
42 in the converting step.

1 34. The computer-readable recording medium of Claim 33,
2 wherein the program counter relative value
3 calculating step includes a lower bit subtracting substep
4 and an upper bit subtracting substep,
5 the lower bit subtracting substep subtracting lower
6 bits of the address of the one of the two specified
7 instructions from lower bits of the address of the other of
8 the two specified instructions, for setting a result of a
9 subtraction as lower bits of the program counter relative
10 value, and sending any carry generated in the subtraction
11 to the upper bit subtracting substep, and
12 the upper bit subtracting substep subtracting upper

13 bits of the address of one of the two specified
14 instructions and any carry received from the lower bit
15 subtracting substep from upper bits of the address of the
16 other of the two specified instructions, and for setting a
17 result of a subtraction as upper bits of the program
18 counter relative value.

1 35. The computer-readable recording medium of Claim 33,
2 wherein the program counter relative value
3 calculating step includes a lower bit subtracting substep
4 and an upper bit subtracting substep,
5 the lower bit subtracting substep subtracting lower
6 bits of the address of one of the two specified
7 instructions from lower bits of the address of the other of
8 the two specified instructions without generating a carry
9 and setting a result of a subtraction as lower bits of the
10 program counter relative value, and
11 the upper bit subtracting substep subtracting upper
12 bits of the address of one of the two specified
13 instructions from upper bits of the address of the other of
14 the two specified instructions, and for setting a result of
15 a subtraction as upper bits of the program counter relative
16 value.

1 36. The computer-readable recording medium of Claim 33,
2 wherein the program counter relative value
3 calculating step subtracts upper bits of an address of one

4 of the two specified instructions from upper bits of an
5 address of the other of the two specified instructions,
6 sets a result of a subtraction as upper bits of the program
7 counter relative value, and sets lower bits of the other of
8 the two specified instructions as lower bits of the program
9 counter relative value.

1 37. A computer-readable recording medium storing an
2 assembler program that generates relocatable code from
3 optimized code that have been generated from an instruction
4 sequence, each address of an instruction in the optimized
5 code having upper bits that indicate a memory address at
6 which a processing packet is stored and lower bits that
7 indicate a position of processing target instruction that
8 is included in the processing packet,

9 the assembler program comprising:

10 a label detecting step for detecting a label in the
11 instruction sequence that should be resolved by a
12 difference in addresses between two specified instructions,
13 and obtaining the addresses of the two specified
14 instructions;

15 a program counter relative value calculating step for
16 calculating a program counter relative value by subtracting
17 an address of one of the two specified instructions from an
18 address of another of the two specified instructions; and

19 a replacing step for replacing the label with the
20 program counter relative value calculated by the program

21 counter relative value calculating step.

1 38. The computer-readable recording medium of Claim 37,
2 wherein the program counter relative value
3 calculating step includes a lower bit subtracting substep
4 and an upper bit subtracting substep,

5 the lower bit subtracting substep subtracting lower
6 bits of the address of the one of the two specified
7 instructions from lower bits of the address of the other of
8 the two specified instructions, for setting a result of a
9 subtraction as lower bits of the program counter relative
10 value, and sending any carry generated in the subtraction
11 to the upper bit subtracting substep, and

12 the upper bit subtracting substep subtracting upper
13 bits of the address of one of the two specified
14 instructions and any carry received from the lower bit
15 subtracting substep from upper bits of the address of the
16 other of the two specified instructions, and for setting a
17 result of a subtraction as upper bits of the program
18 counter relative value.

1 39. The computer-readable recording medium of Claim 37,
2 wherein the program counter relative value
3 calculating step includes a lower bit subtracting substep
4 and an upper bit subtracting substep,

5 the lower bit subtracting substep subtracting lower
6 bits of the address of one of the two specified

COPIED FROM THE

instructions from lower bits of the address of the other of the two specified instructions without generating a carry and setting a result of a subtraction as lower bits of the program counter relative value, and

the upper bit subtracting substep subtracting upper bits of the address of one of the two specified instructions from upper bits of the address of the other of the two specified instructions, and for setting a result of a subtraction as upper bits of the program counter relative value.

40. The computer-readable recording medium of Claim 37,

wherein the program counter relative value calculating step subtracts upper bits of an address of one of the two specified instructions from upper bits of an address of the other of the two specified instructions, sets a result of a subtraction as upper bits of the program counter relative value, and sets lower bits of the other of the two specified instructions as lower bits of the program counter relative value.

41. A computer-readable recording medium storing a linker program that generates object code from relocatable code that has been generated from an instruction sequence, each address of an instruction in the optimized code having upper bits that indicate a memory address at which a processing packet is stored and lower bits that indicate a

7 position of processing target instruction that is included
8 in the processing packet,

9 the linker program comprising:

10 a relocation information detecting step for detecting
11 a label in the relocatable code that should be resolved by
12 a difference in addresses between two specified
13 instructions, and obtaining the addresses of the two
14 specified instructions;

15 a program counter relative value calculating step for
16 calculating a program counter relative value by subtracting
17 an address of one of the two specified instructions from an
18 address of another of the two specified instructions; and

19 a replacing step for replacing the label with the
20 program counter relative value calculated by the program
21 counter relative value calculating step.

1 42. The computer-readable recording medium of Claim 41,

2 wherein the program counter relative value
3 calculating step includes a lower bit subtracting substep
4 and an upper bit subtracting substep,

5 the lower bit subtracting substep subtracting lower
6 bits of the address of the one of the two specified
7 instructions from lower bits of the address of the other of
8 the two specified instructions, for setting a result of a
9 subtraction as lower bits of the program counter relative
10 value, and sending any carry generated in the subtraction
11 to the upper bit subtracting substep, and

3 calculating step subtracts upper bits of an address of one
4 of the two specified instructions from upper bits of an
5 address of the other of the two specified instructions,
6 sets a result of a subtraction as upper bits of the program
7 counter relative value, and sets lower bits of the other of
8 the two specified instructions as lower bits of the program
9 counter relative value.

1 45. A computer-readable recording medium storing a compiler
2 program that generates an instruction sequence from source
3 code,

4 the compiler program generating a program counter
5 relative value calculating instruction that is executed by
6 a processor, the program counter relative value calculating
7 instruction being an instruction that performs a
8 calculation using a first value and a program counter
9 relative value and uses a result of the calculation to
10 update the first value, the first value being one of

11 (a) a value of a program counter stored in a
12 register, and

13 (b) the value stored in a program counter of the
14 processor,

15 wherein upper bits of the first value indicate a
16 memory address at which a processing packet is stored, and
17 lower bits of the first value of the program counter
18 indicate a processing target instruction that is included
19 in the processing packet.

46. The computer-readable recording medium of Claim 45, wherein the processor includes a lower bit calculating unit and an upper bit calculating unit, the program counter relative value calculating instruction having the lower bit calculating unit perform a lower bit calculation and the upper bit calculating unit perform an upper bit calculation, the lower bit calculation being an addition using lower bits of the first value and lower bits of the value of the program counter relative value, where a result of the lower bit calculation is set as the lower bits of the first value and any generated carry is sent to the upper bit calculating unit, and the upper bit calculation being an addition using upper bits of the first value, upper bits of the value of the program counter relative value and any carry received from the lower bit calculating unit, where a result of the upper bit calculation is set as the upper bits of the first value.

47. The computer-readable recording medium of Claim 45,
wherein the processor includes a lower bit
calculating unit and an upper bit calculating unit,
the program counter relative value calculating
instruction having the lower bit calculating unit perform a
lower bit calculation and the upper bit calculating unit

7

8

14

1

4

9

A.D.D B³
Ad C²